Rec'd PCT/PTO 03 FEB 2005

(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
28 March 2002 (28.03.2002)

PCT

(10) International Publication Number
WO 02/25880 A2

(51) International Patent Classification[7]: H04L 12/64

(21) International Application Number: PCT/CA01/01313

(22) International Filing Date: 17 September 2001 (17.09.2001)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
09/664,712 19 September 2000 (19.09.2000) US

(71) Applicant (*for all designated States except US*): **NORTEL NETWORKS LIMITED** [CA/CA]; 2351 Boulevard Alfred-Nobel, St. Laurent, Quebec H4S 2A9 (CA).

(72) Inventor; and
(75) Inventor/Applicant (*for US only*): **BLOUIN, François, F.** [CA/CA]; 29 de la Comete, Hull, Quebec J9A 2Y5 (CA).

(74) Agents: **HARRIS, John, D.** et al.; Gowling Lafleur Henderson LLP, Suite 2600, 160 Elgin Street, Ottawa, Ontario K1P 1C3 (CA).

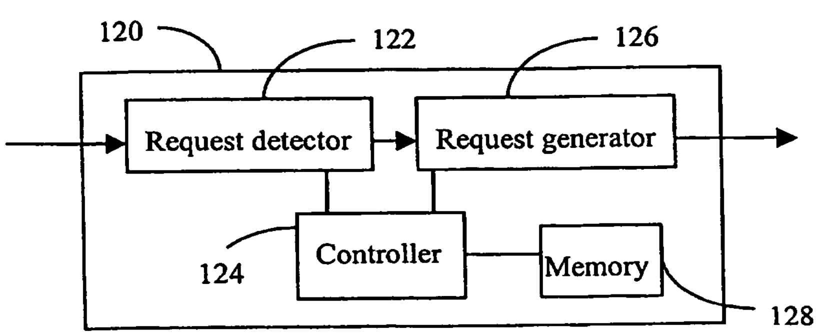
(81) Designated States (*national*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, MZ, NO, NZ, PH, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, US, UZ, VN, YU, ZA, ZW.

(84) Designated States (*regional*): ARIPO patent (GH, GM, KE, LS, MW, MZ, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE, TR), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

(54) Title: DYNAMIC TCP CONFIGURATION FOR LOW LATENCY VOICE/DATA TRAFFIC



(57) Abstract: There are many circumstances in which delay sensitive traffic and a delay insensitive traffic share an access to a telecommunications network. TCP/IP is well suited for proper delivery of the delay insensitive traffic whereas the delay sensitive traffic uses UDP/IP protocol. When a delay sensitive traffic in UDP/IP packets is in progress, ACK delay parameter in TCP/IP protocol can be adjusted so that UDP packets can be inserted among TCP packets to reduce collision and end-to-end delay of UDP packets.

WO 02/25880 A2

Dynamic TCP Configuration for Low Latency Voice/Data Traffic

Field of Invention

The invention generally resides in the field of handling multimedia traffic in a communications network domain. In particular, the invention is concerned with voice and data traffic in TCP/IP sharing an access to a network.

Background of Invention

A communications network handles data traffic among many computers, printers, servers, and other end devices (data terminals). The network can be in the form of an Ethernet, token ring, switching network or other. Whatever the type of network is used, each of these end devices has a dedicated access (sometimes called a network drop or simply a drop) connecting to the network. In a typical office environment, telephone sets are also essential equipment and generally form a separate telephone network among them. Two separate networks require additional cost for installation, maintenance, etc and also lead to an inefficient use of network resources such as bandwidth. Telephone sets, however, can connect to and share the communications network with data terminals. This will do away with the telephone network, resulting an economical benefit. Each telephone set can connect to the communications network by a dedicated drop like other end devices but as an office generally requires a data terminal and a voice terminal (e.g., telephone set) at close proximity to one another, it would be beneficial if both terminals can share one drop. When designing a shared drop, it should be borne in mind that a voice call cannot tolerate a large and variable delay. A switch on a shared drop can control the traffic to/from a data terminal or voice terminal, giving the voice traffic a priority over the data traffic. Switches, however, are an expensive hardware and providing a switch on each shared drop increases the cost even more and is not very practical. Hubs also allow different traffic to share an access but they have a problem of collision that must be addressed.

TCP/IP and UDP/IP are widely popular protocols for transferring data in packets through networks. TCP (Transmission Control Protocol) provides a reliable stream delivery and virtual connection services to applications through the

use of sequenced acknowledgment with retransmission of packets when necessary. Delay insensitive traffic such as file transfer, transactions on the web etc. use mainly TCP/IP. UDP (User Datagram Protocol), on the other hand, provides a simple, but unreliable message service for transaction-oriented services with minimum of protocol mechanism. Each UDP header carries both a source port identifier and destination port identifier, allowing high-level protocols to target specific applications and services among host. UDP packets are suitable for sending delay sensitive traffic such as voice etc.

Summary of Invention

The invention addresses problems associated with provisioning a data terminal and a voice terminal on a shared drop.. In one aspect, the invention resides in TCP/IP and UDP/IP environment and it configures dynamically TCP settings to provide low latency voice delay by prioritizing voice over delay insensitive data packet. In a further aspect, the invention contemplates the use of a 3-port hub on the shared drop.

In accordance with one aspect, the present invention relates to a method of sharing an access to a communications network by a delay sensitive traffic in UDP/IP and a delay insensitive traffic in TCP/IP. The method includes steps of adjusting an ACK delay parameter for the delay insensitive traffic by a desired amount in response to a request for a delay sensitive traffic and sending the delay sensitive traffic in a series of UDP packets through the access. The method further includes a step of sending an acknowledgment of a reception of delay insensitive traffic through the access at the adjusted ACK delay parameter.

In accordance with another aspect, the invention is directed to a communications module for permitting delay sensitive UDP/IP traffic and delay insensitive TCP/IP traffic to share an access to a communications network. The communication module comprises a detection unit for detecting a request for a delay sensitive UDP traffic. It also includes a request generating unit for generating a request signal in response to the request and sending the request signal to a TCP/IP unit of the delay insensitive TCP/IP traffic. The request signal contains an instruction for the TCP/IP unit to adjust the ACK delay parameter.

In accordance with yet a further aspect, the invention is directed to a hub in an access to a communications network for allowing a delay sensitive UDP/IP traffic and a delay insensitive TCP/IP traffic to share the access. The hub comprises a first port for exchanging the delay sensitive UDP/IP traffic, a second port for exchanging the delay insensitive TCP/IP traffic and a third port for exchanging both delay sensitive UDP/IP traffic and delay insensitive TCP/IP traffic with the communications network. The hub further includes a detection unit for detecting a request for a delay sensitive UDP traffic, and a request generating unit for generating a request signal in response to the request and for sending the request signal to a TCP/IP unit for the delay insensitive TCP/IP traffic. The request signal contains an instruction for the TCP/IP unit to adjust the ACK delay parameter.

In accordance with a further aspect, the invention is directed to a telephone set for sharing an access to a communications network with an end device which generates a delay insensitive TCP/IP traffic. The telephone set includes a request detection unit for detecting a request for delay sensitive UDP traffic and in response to the request. The set further includes a request generating unit for generating a request signal and sending the request signal to a TCP/IP unit for the delay insensitive TCP/IP traffic, the request containing an instruction for the TCP/IP unit to adjust the ACK delay parameter, and a DSP module for generating a series of UDP packets in response to an audio input of the telephone set and sending the series of UDP packets through the access.

In accordance with another aspect, the invention is directed to a computer terminal operating in TCP/IP for sharing an access to a communications network with a telephone set which generates a delay sensitive UDP/IP traffic. The computer terminal includes a request detector for detecting a request for delay sensitive UDP/IP traffic and a request generating unit for generating a request signal, the request signal containing an instruction for the computer terminal to adjust the ACK delay parameter. The terminal further includes a TCP/IP unit for adjusting the ACK delay parameter in response to the request signal.

In accordance with a further aspect, the invention is directed to a first computer terminal operating in TCP/IP for sharing an access to a communications network with a second computer terminal which generates a delay sensitive voice

traffic in the form of UDP/IP packets. The first computer terminal includes a request detector for detecting a request for delay sensitive UDP/IP traffic from the second computer and a request generating unit for generating a request signal, the request signal containing an instruction for the first computer terminal to adjust the ACK delay parameter. The first terminal further includes a TCP/IP unit for adjusting the ACK delay parameter in response to the request signal.

Brief Description of Accompanying Drawings

Figure 1 shows a communications environment in which the present invention finds its applications.

Figure 2 shows the operation of a switch.

Figure 3 shows the operation of a hub.

Figure 4 depicts an Ethernet communications network connecting variety of end terminals with shared drops.

Figure 5 illustrates voice traffic in UDP packets on an access to an Ethernet network.

Figure 6 illustrates data traffic in TCP packets on an access to an Ethernet network.

Figure 7 shows data traffic in TCP packets on an access to an Ethernet network when the piggyback feature is used.

Figure 8 is a timing diagram and call process chart showing functional phases of one embodiment of the invention.

Figure 9 is a flow chart of processing a voice call request and adjusting the ACK delay parameter.

Figure 10 illustrates voice in UDP packets and data traffic in TCP packets sharing an access to an Ethernet network in accordance with one embodiment of the invention.

Figure 11 is a block diagram of a hardware configuration of a processing module according to an embodiment of the present invention.

Figure 12 is a block diagram of a few embodiments of the present invention, showing possible locations of the processing module.

Figure 13 is a block diagram of a further embodiments of the present invention, showing a pair of computers having voice capabilities.

Figures 14, 15, 16, 17, 18 and 19 show simulation results at either the standard (default) setting of 10 ms or the increased delay setting of 200 ms.

Detailed Description of Preferred Embodiments of Invention

Referring to Figure 1, a communications network 10 includes a plurality of drops connecting telephone sets 12, computers 14, servers 16, printers 18 and other end devices. It also has a connection to other networks 20 by way of a gateway 22. In the Figure a telephone set and a computer share a single drop 24 connecting to the network by way of a module 26. As described earlier, the module can take a form of a switch. Switches are a layer-2 device (e.g., IP layer). As shown in Figure 2, a switch 30 includes a scheduler 32 and buffers 34. Traffic from a plurality of input ports 36 is stored in the buffer and the scheduler controls a proper order of traffic when outputting from an output port 38. In the present circumstance, a switch requires only three ports, one to a telephone set, another to a data terminal and the third being an uplink to the network. As mentioned above, switches are a costly hardware, even when it only deals with three ports.

Hubs, on the other hand, are a layer-1 device. They are also known as a repeater and operate in half duplex, that is to say, traffic only travels in both directions but in one direction at a time. Figure 3 shows an n-port hub 40. When a hub receives traffic at one port 42, it copies to all the remaining ports 44. A device of layer-2 or higher connected to each remaining port decides whether to accept or ignore the copied traffic. Hubs are simpler and less costly than switches but hubs encounter collisions which need to be addressed. When a hub receives incoming traffic at two or more ports at a same time, a collision occurs, resulting in corrupted or dropped data from the traffic. In an Ethernet network, when a transmitting end device senses a collision, it backs off from transmitting and waits for a certain period of time before it attempts a retransmit. All the transmitting end devices wait for different periods to avoid repeated collisions. Different types of network use different mechanisms for avoiding or eliminating collisions. The present invention proposes mechanisms which lessen the problem of collisions when delay sensitive and delay insensitive traffic share an access to a network.

In accordance with one preferred embodiment, the invention uses a 3-port hub as a module shown in Figure 1. In Figure 4, therefore, a computer 50 is

exchanging data with a FTP server, web server or the like 52 through a network 54, using TCP packets. A telephone set 56 includes a feature which permits to transmit and receive voice traffic in UDP packets. Such telephone sets are available now on the market as IP phones. A hub 58 in cooperation with a call server 60 permits the sharing of a drop 62 by the computer and the telephone set. The network in this embodiment is an Ethernet network but other types of network e.g., PSTN, ATM, Token Ring, FDDI, PPP etc. are also possible.

Voice on a telephone set is generally sampled for 10-30 ms (milliseconds) and another 10-15 ms is required to processing, packetization etc of sampled data. This would produce over 20 samples/second (sampling rate). With some overhead etc., a sampling duration of 30 ms produces a UDP packet of 280 bytes long at each sampling. The voice packets or voice UDP packets can be processed and compressed to suite for transport over a specific network. Each UDP packet therefore carries each sampled voice datum. On a 10baseT Ethernet network, for example, each voice UDP packet (280 bytes in size) would have a duration of about 224 μs (microseconds). Ideally voice UDP packets are uniformly transmitted through the network or uniformly received by the destination telephone set at the same sampling rate (e.g., 45 ms apart at 22 samples /second) and without a large end-to-end delay.

Figure 5 shows voice traffic of the example described above on a 10BaseT Ethernet, the horizontal axis showing the time t (or translated into number of bytes). The 10BaseT Ethernet transfers data through the network at a rate of 10 million bits per second (Mb/s) (=1.25 Mbytes/s). The Figure shows a plurality of voice UDP packets 72, each of which is 224 μs (280 bytes) in duration. Two adjacent UDP packets are at least 45 ms (56250 bytes) apart.

Figure 6, on the other hand, shows data traffic on a 10BaseT Ethernet network. The traffic may be of an FTP transaction between the computer and a server, printing a file off a printer, or an HTTP transaction on the web etc. The transaction is in a plurality of TCP packets, each of which can be theoretically up to 65,000 bytes long. An Ethernet limits the maximum size of packets to 1,500 bytes (equivalent of 1.2 ms on 10baseT). TCP protocol features an ACK mechanism for TCP packets to ensure proper delivery of packets. The end devices use sequence numbers indicating sent or received data for the ACK mechanism

and the sliding window mechanism for flow control. When an end device receives TCP packets, it sends an ACK signal to the transmitting end device, informing the reception of packets by using the sequence numbers. The transmitting end device waits for the ACK signal before it transmits the next TCP packet. A transmission may contain more than one packet and a group of packets in one transmission is called a segment. Both end devices use the sliding window mechanism to inform each other how large the next segment can be. TCP also provides a time-out period of e.g., 1-2 seconds. If an end device does not receive an expected ACK signal and the time-out period elapses, the end device assumes that the transmitted packets lost and retransmits the lost TCP packet or packets. In TCP, each end device is designed to send an ACK signal to the transmitting end device within an ACK delay parameter of the reception of packets. Each end device can set its own ACK delay parameter. The ACK delay parameter obviously must be less than the time-out period or the transmitting end device will perform a retransmit.

The ACK delay parameters can be set in terms of a time on a clock or by the number of segments transmitted. By clock, the delay can be set by a time period of e.g., from 1 ms to 200 ms. It can also be set by the segment so that in one example it can be set to acknowledge every other segment, taking into consideration of the time-out period.

A receiving end device can send an independent ACK signal according to the ACK delay parameters. On the other hand, when two end devices are exchanging data (data going in both directions), an end device can piggyback an ACK signal onto the next TCP packet or packets it sends to the other end device, observing the ACK delay parameter.

Figure 6 therefore indicates an FTP transaction in TCP packets 82 with ACK signals 84. The TCP packets are variable in size and travel in one direction, while the ACK signals in the opposite direction. Because Ethernet networks are bus network, the figures are considered to depict the periods of the bus being driven by one end device. Some example values of TCP packet's sizes are given in the figure, but generally much longer than voice UDP packets discussed above. A period between an ACK signal and the next TCP packet depend on the availability of packets to be sent and processing time at the transmitting end device. In the Figure, a data terminal for example is set to send an ACK signal 10

ms after the reception of a TCP packet. At 86, for some reason 10 ms elapsed and no ACK signal was sent. This happened, for example, when another end device attempted to drive the bus coincidentally with the ACK signal and caused a collision. A retransmit 88 of the presumed lost TCP packet followed.

Figure 7 shows traffic when two end devices are using the piggyback feature by which an ACK signal is concatenated to a TCP packet being transmitted. In this example, each end device alternately drives the bus to send its TCP packet. One end device has the ACK delay parameter set at 10 ms and the other at 15 ms.

UDP packets are continuously sent and there is no ACK mechanism in transporting data. Voice UDP packets should travel through the network with as little delays and as uniformly apart from one another as possible. The end-to-end delay of more than 150 ms is considered unacceptable for voice data. If some voice UDP packets experience more than 150 ms of delay, voice would suffer from noise, distortion, voice clipping etc. TCP packets of file transfer etc., on the other hand, are generally large in size but they are not sensitive to delay. Proper delivery is more important in those situations and the ACK mechanism ensures it. As mentioned above, TCP packets prompt ACK signals within an ACK delay parameter, e.g., 10 ms after reception or every other segments received for the transmitting end device to continue further transmission.

The shared access to the network becomes available when no traffic exists on the access. When the shared access becomes available, both packets, UDP and TCP, compete equally for access. TCP packets are, however, generally longer and variable in size than voice UDP packets which are also spaced further apart. In competing with TCP packets, therefore, voice UDP packets experience variable and often unacceptably large delays. Often, once a TCP transaction had started, the voice UDP packet has very little chance to seize the shared access until the TCP finish the transaction, if the ACK delay parameter is set at e.g., 10 ms (normally a default value).

TCP has many parameters that can be tuned to achieve different level of transmission and bandwidth utilization. The default windows setting (10 ms delay parameter setting) is for the fastest transmission and no bandwidth sharing with other devices on the same collision domain e.g., the same Ethernet. By modifying

dynamically TCP setting when voice calls are in progress, TCP traffic can be slowed down in order to give up some bandwidth of the shared access to delay sensitive traffic such as voice traffic. For example as one embodiment, by modifying the TCP ACK delay parameter, priority can be given to voice UDP packets while TCP data traffic throughput is reduced.

According to one embodiment, the TCP ACK delay parameter can be increased from the typical 10 ms to 200 ms at an end device which shares an access with a voice terminal. This will leave a gap of e.g., 200 ms before the end device sends an ACK signal, thus making the other end device to slow down the transmission of further packets. Voice UDP packets can be inserted into this gap. The new setting of ACK delay parameter can be empirically determined or adjusted for desired performance. The voice traffic should not suffer an end-to-end-delay of more than 150 ms. It can intuitively be predicted that as the ACK delay parameter is increased, the end-to-end delay performance of voice traffic should improve to a certain level where the performance should level out while the throughput of TCP packets continues to decrease with the increased setting of ACK delay parameter. The TCP time-out setting will determine the maximum setting of ACK delay parameter. When the voice call terminates, the ACK delay parameter is reset to a default value. The ACK delay parameters are dynamically set, that is to say, whenever a voice call is requested, the data terminal is informed of the voice call request and instructed to set the ACK delay parameter to a new increased setting. Each voice terminal on the network may have a feature that informs the sharing data terminal of a voice call request which is generally indicated by a telephone receiver going off-hook or some such signal. It should be noted that while an IP phone or some such voice terminal has been described thus far as a voice terminal in the embodiments, a computer terminal with a proper software is able to function as a voice terminal. In another embodiment, the network has a call server to which voice terminals on the network send a voice call request prior to initiate the voice call. Upon reception of the call request, the call server uses some criteria e.g., bandwidth, capability of voice terminal etc. and determines if a voice call can be accepted. If accepted, the call server informs the voice terminal which in turn informs its access sharing data terminal to set a new ACK delay parameter setting because a voice call will start soon. The data

terminal performs or continues to perform a TCP transaction under the new setting while the voice call is in progress. Upon termination of the voice call, the data terminal reset the ACK delay parameter.

Figure 8 shows call flows at some phases of operation of one preferred embodiment of the invention.

At phase (1): An IP phone informs a call server that it desires to make a phone call to another IP phone, by sending a call request.

At phase (2): The call server accepts or denies the call request, based on criteria, e.g., the availability of bandwidth, etc.

At phase (3): If the call is accepted, the IP phone sends a message to its neighbor end device e.g., a PC (on the same shared hub) to notify the PC that a voice call will start soon and its TCP settings should be configured to allow voice packets to be sent out with minimum delay.

At phase (4): Voice call is established between two IP phones on the network. Voice UDP packets are transmitted at an interval shown by 90

At phase (5): PC data transfer is executed while the voice call is in progress. The new delay parameter setting is shown by 92. The new TCP settings allow for both voice and data traffic to be transferred on the shared hub while maintaining adequate voice end-to-end delay of less than 100 or 150 ms. The data traffic between the PC and a web server or file server continues with an increased delay.

Figure 9 shows a flow chart of the operation shown in Figure 8. If there is either a voice call request or a voice call termination request is received at 100, either request instructs a TCP unit located at a computer terminal to adjust the ACL delay parameter at 102. In parallel at 104, an IP phone starts or receives voice UDP packets. Meanwhile if TCP packets are received at 106, the TCP unit sends ACK in accordance with the adjusted ACK delay parameter at 108. When a voice termination request is received at 110, it is detected and the ACK delay parameter is adjusted again, likely back to the default setting.

Figure 10 shows the sharing of the access by both voice UDP packets and TCP packets according to one embodiment of the invention. In the Figure, voice UDP packets are permitted to travel at 45 ms interval, while TCP packets experience differing and generally longer delays.

Referring to Figure 11, in accordance with one embodiment, the invention can be realized in a processing module 120. As shown in Figure 12, the processing module could be located at an IP phone or a computer terminal which shares the access with the IP phone or a hub. On the other hand, it could be located at the call server located at another location on the network as shown in Figure 4. Referring to Figure 11, the processing module contains a request detector unit 122 which detects a telephone set going off-hook or a similar signal, indicating that a delay sensitive UDP (a voice call) is requested. The request detector unit in one embodiment detects also presence or termination of voice call. Upon detection of the off-hook condition and under control of a controller 124, request generator 126 generates a request for UDP traffic and sends the request to the computer terminal, instructing to adjust the ACK delay parameter of a TCP/IP module to a new setting. The controller refers to a memory 128 for stored information such as addresses of a calling and called party and other relevant data to the hub, computer terminal or call server. When the processing module is located at the computer terminal, the processing unit directly instructs the TCP/IP module of the computer terminal to adjust the ACK delay parameter. When the termination of voice call (UDP packets) is detected, the controller indicates such condition to the TCP/IP module which then resets the ACK delay parameter to the initial value.

Figure 12 shows possible locations of processing module 130. Referring to Figure 12, an IP phone 132 contains a DSP module 134 for processing and packetizing audio input to generate series of UDP packets. A computer terminal 136 requires a TCP/IP processing unit 138 in addition to other essential component such as computer processor 140. A processing module can be located at a hub 142 which contains a necessary hub function units 144. Figure 13 shows another embodiment in which a pair of computers shares an access. Both or one of the computers contains a voice processing software for generating voice UDP packets, thus functioning as a voice terminal.

Figures 14-19 show simulation results at either the standard (default) setting of 10 ms or the increased setting of 200 ms. The horizontal axes are in arbitrary time scale.

Figure 14 includes two graphical representations in the same time scale of an end-to-end delay (in second) and counts of collision/transmission attempt under standard TCP settings.

Figure 15 includes two graphical representations in the same time scale of an end-to-end delay (in second) and counts of collision/transmission attempt under a new ACK delay setting of 200 ms.

As can be seen in Figure 14, the upper graph shows that there are many voice packets which experienced over 0.1 second (100 ms) delay and there are substantial packets with more than 150 ms delay. In the lower graph, many TCP transmission attempts experienced collisions. For example, at about 6m30, there are about 150 attempts and 125 collisions. This translates to only about 16% (=25/150) success. In Figure 15, there is only one instance in which a packet experienced a delay of a little over 0.08 seconds (80 ms), all the remaining voice packets being delayed less than 80 ms. As for collision, the lower graph shows that almost 50% success rate throughout the voice call.

Figures 16 and 17 each includes two graphical representations of voice delay distribution respectively under standard TCP settings and under a new ACK delay setting of 200 ms. Figures 16 and 17 used same data as those used in Figures 14 and 15 but show them differently.

Therefore, Figures 16 and 17 correspond to Figures 14 and 15 respectively. It is clear that there are many voice packets with delay of more than 150 ms under the standard setting whereas almost all voice packets have less than 80 ms delay under the new setting.

Figures 18 and 19 show groups of graphical representations of voice sharing with a 10 M bytes FTP transaction under the TCP standard and under the new setting of 200ms ACK delay parameter respectively.

In Figure 18, there are many voice packets with more than 150 ms delay. Transmission attempts vs. collisions by voice packets are shown in the second graph while those by data packets from a computer are in the third graph. Both graphs indicate rather high rates of collisions (low rates of successes). The last graph shows that the hub is in constant utilization. Turning to Figure 19, all the corresponding results under the new setting indicate marked improvement on all the performance parameters.

Although illustrative embodiments of the invention have been shown and described, other modifications, changes, and substitutions are intended in the foregoing disclosure. Accordingly, it is appropriate that the appended claims be construed broadly and in a manner consistent with the scope of the invention.

What is claimed is:

1. A method of sharing an access to a communications network by a delay sensitive traffic transported in a first protocol and a delay insensitive traffic transported in a second protocol, comprising steps of:

detecting a request for the delay sensitive traffic;

adjusting an ACK delay parameter for the delay insensitive traffic by a desired amount in response to the request;

sending the delay sensitive traffic in a series of first protocol packets through the access, and

sending an acknowledgment of a reception of delay insensitive traffic through the access at the adjusted ACK delay parameter.

2. The method of sharing an access to a communications network according to claim 1, wherein the first protocol is UDP/IP, the method further comprising a step of:

sending the delay sensitive traffic in a series of UDP packets through the access at substantially a regular interval.

3. The method of sharing an access to a communications network according to claim 2, wherein the second protocol is TCP/IP, the method further comprising a step of:

sending the delay insensitive traffic in TCP packets through the access at the adjusted ACK delay parameter.

4. The method of sharing an access to a communications network according to claim 3 wherein the access comprises a 3-port hub, one of three port being connected to the network, the method further comprising steps of sending the delay sensitive traffic through one remaining port of the hub and sending the delay insensitive traffic through another remaining port of the hub.

5. The method of sharing an access to a communications network according to claim 1, wherein the second protocol is TCP/IP, the method further comprising a step of:

adjusting the ACK delay parameter in terms of either clock or the number of segments.

6. The method of sharing an access to a communications network according to claim 5, further comprising a step of:

adjusting the ACK delay parameter by setting the clock at 150 ms or over.

7. The method of sharing an access to a communications network according to claim 6, further comprising a step of:

sending an acknowledgment piggybacked on a TCP packet of the delay insensitive traffic at the adjusted ACK delay parameter.

8. The method of sharing an access to a communications network according to claim 3, further comprising a step of:

resetting the adjusted ACK delay parameter at the end of the delay sensitive UDP/IP traffic.

9. The method of sharing an access to a communications network according to claim 5, wherein the first protocol is UDP/IP, the method further comprising a step of:

resetting the adjusted ACK delay parameter at the end of the delay sensitive UDP/IP traffic.

10. A communications module for permitting delay sensitive traffic transported in a first protocol and delay insensitive traffic transported in a second protocol to share an access to a communications network, comprising:

a detection unit for detecting a request for a delay sensitive traffic, and

a request generating unit for generating a request signal in response to the request and sending the request signal to a processing unit of the delay insensitive

traffic, the request signal containing an instruction for the processing unit to adjust the ACK delay parameter.

11. A communications module for permitting delay sensitive traffic and delay insensitive traffic to share an access to a communications network according to claim 10, further comprising:

a controller connected to the detection unit and the request generating unit for generating a request signal for delay sensitive traffic, which request signal further contains a called number and a calling number of the delay sensitive traffic.

12. A communications module for permitting delay sensitive traffic and delay insensitive traffic to share an access to a communications network according to claim 11, wherein the first protocol is UDP/IP and the controller further generates a signal indicating the termination of the delay sensitive UDP/IP traffic, the signal containing an instruction for the processing unit to reset the ACK delay parameter.

13. A communications module for permitting delay sensitive traffic and delay insensitive traffic to share an access to a communications network according to claim 12, wherein the second protocol is TCP/IP and the processing unit is a TCP/IP unit.

14. A hub in an access to a communications network for allowing a delay sensitive traffic transported in a first protocol and a delay insensitive traffic transported in a second protocol to share the access, comprising:

a first port for exchanging the delay sensitive traffic, a second port for exchanging the delay insensitive traffic and a third port for exchanging both delay sensitive traffic and delay insensitive traffic with the communications network;

a detection unit for detecting a request for a delay sensitive traffic, and

a request generating unit for generating a request signal in response to the request and sending the request signal to a processing unit of the delay insensitive traffic, the request signal containing an instruction for the processing unit to adjust the ACK delay parameter.

15. The hub in an access to a communications network for allowing a delay sensitive traffic and a delay insensitive traffic to share the access according to claim 14, wherein the first protocol is UDP/IP, the hub further comprising:

a detection unit for detecting the termination of the delay sensitive UDP traffic, and

a controller for generating a signal indicating the termination of the delay sensitive UDP traffic and instructing the processing unit to reset the ACK delay parameter.

16. The hub in an access to a communications network for allowing a delay sensitive traffic and a delay insensitive traffic to share the access according to claim 15, wherein the second protocol is TCP/IP and the protocol unit is a TCP/IP unit.

17. A terminal device of a delay sensitive traffic for sharing an access to a communications network with a data end device which generates a delay insensitive traffic, comprising:

a request detection unit for detecting a request for a delay sensitive traffic;

a request generating unit for generating a request signal in response to the request and sending the request signal to a processing unit of the delay insensitive traffic, the request containing an instruction for the processing unit to adjust the ACK delay parameter, and

a DSP module for generating a series of packets of the delay sensitive traffic in response to an input signal and sending the series of packets through the access.

18. The terminal device according to claim 17, further comprising:

the request detection unit for detecting the termination of the delay sensitive traffic, and

a controller for generating a signal indicating the termination of the delay sensitive traffic and instructing the processing unit to reset the ACK delay parameter.

19. The terminal device according to claim 18 wherein the delay sensitive traffic is voice traffic and is in a series of UDP/IP packets.

20. The terminal device according to claim 19 wherein the delay insensitive traffic is in a series of TCP/IP packets and the processing unit is a TCP/IP unit.

21. A computer terminal of a delay insensitive traffic for sharing an access to a communications network with a terminal device which generates a delay sensitive traffic, comprising:

a request detector for detecting a request for delay sensitive traffic;

a request generating unit for generating a request signal, the request signal containing an instruction for the computer terminal to adjust the ACK delay parameter, and

a processing unit for adjusting the ACK delay parameter in response to the request signal.

22. The computer terminal according to claim 21 wherein the delay sensitive traffic is in UDP/IP, the delay insensitive traffic is in TCP/IP and the processing unit is a TCP/IP unit, the computer terminal further comprising:

the request detector for detecting the termination of the delay sensitive UDP traffic, and

a controller for generating a signal indicating the termination of the delay sensitive UDP traffic and instructing the TCP/IP unit to reset the ACK delay parameter.

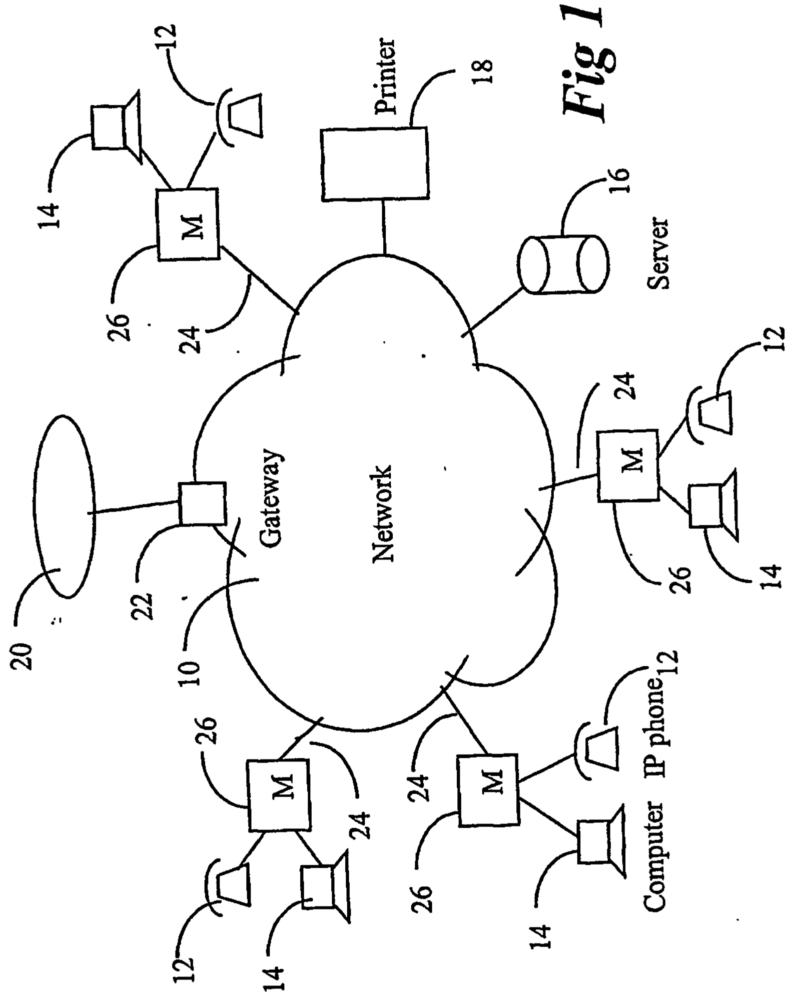

Printer
Server
Gateway
Network
Computer
IP phone
M
10
12
14
16
18
20
22
24
26

Fig 1

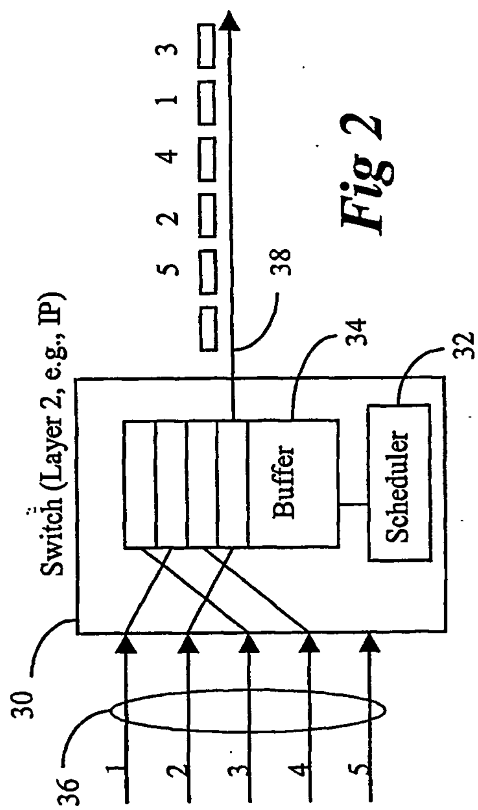

Switch (Layer 2, e.g., IP)
Buffer
Scheduler
30
32
34
36
38
1
2
3
4
5
5 2 4 1 3

Fig 2

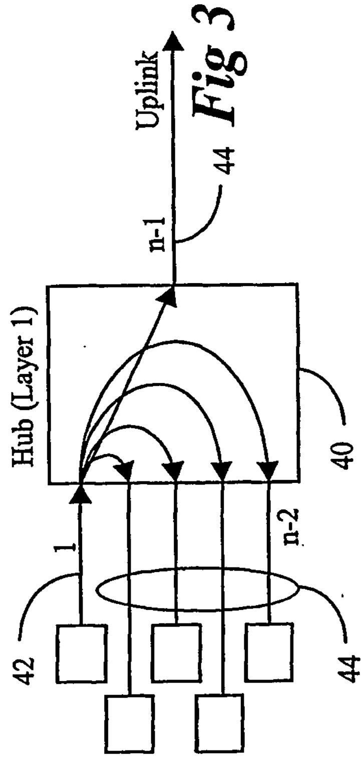

Hub (Layer 1)
Uplink
n-1
n-2
1
40
42
44
44

Fig 3

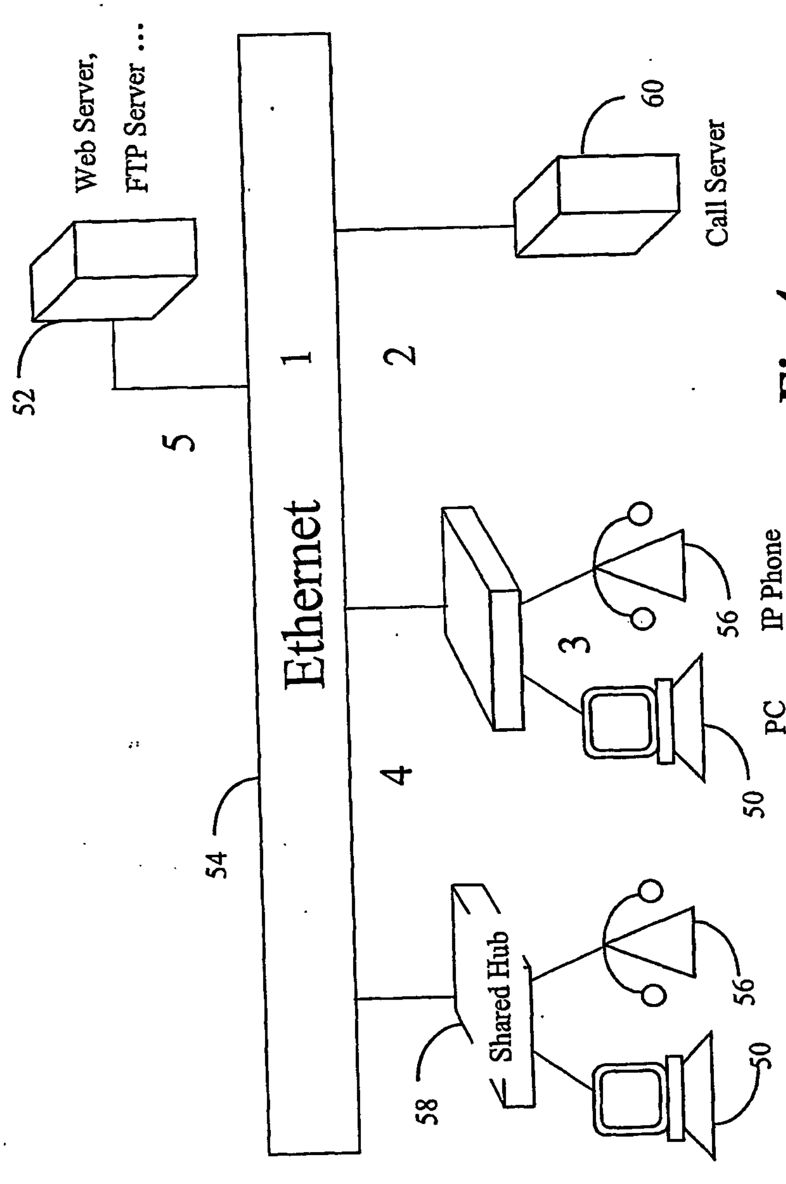

Web Server,
FTP Server ...
52
5
1
2
Ethernet
60
Call Server
3
4
54
56
IP Phone
PC
50
Shared Hub
58
56
50

Fig 4

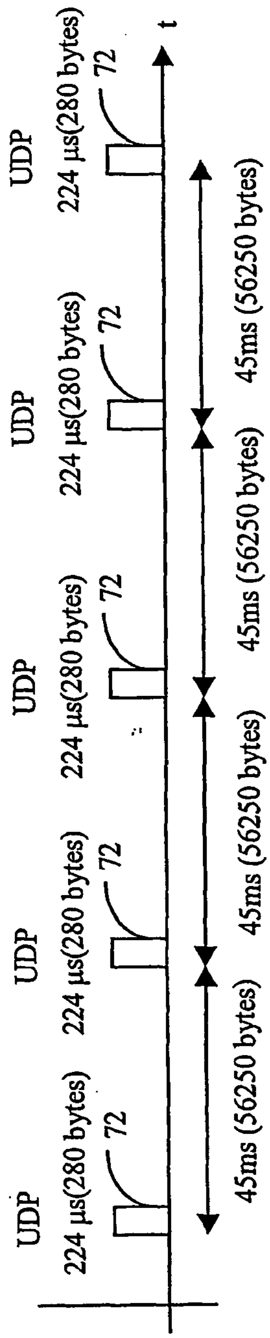


Fig 5

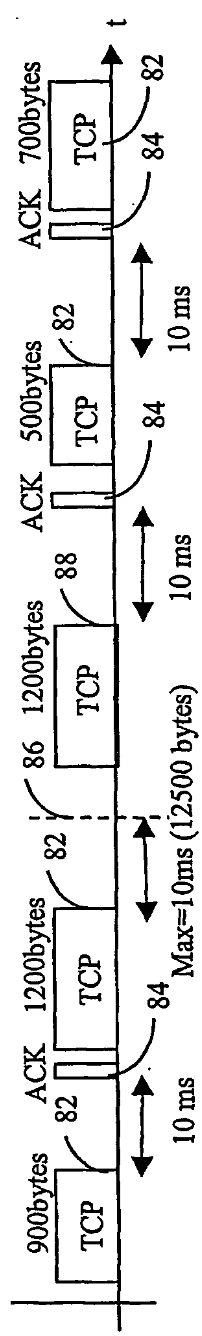


Fig 6

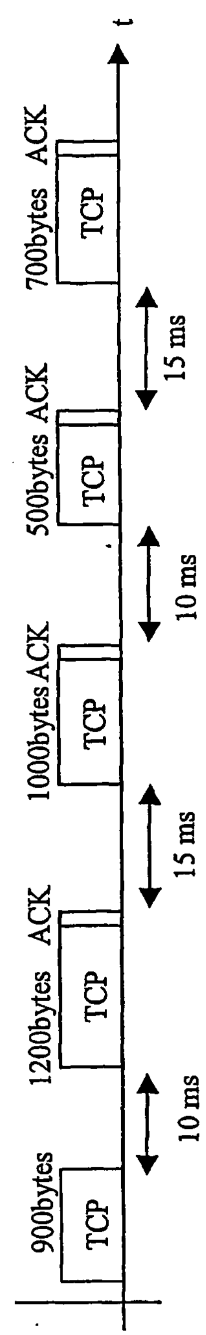


Fig7

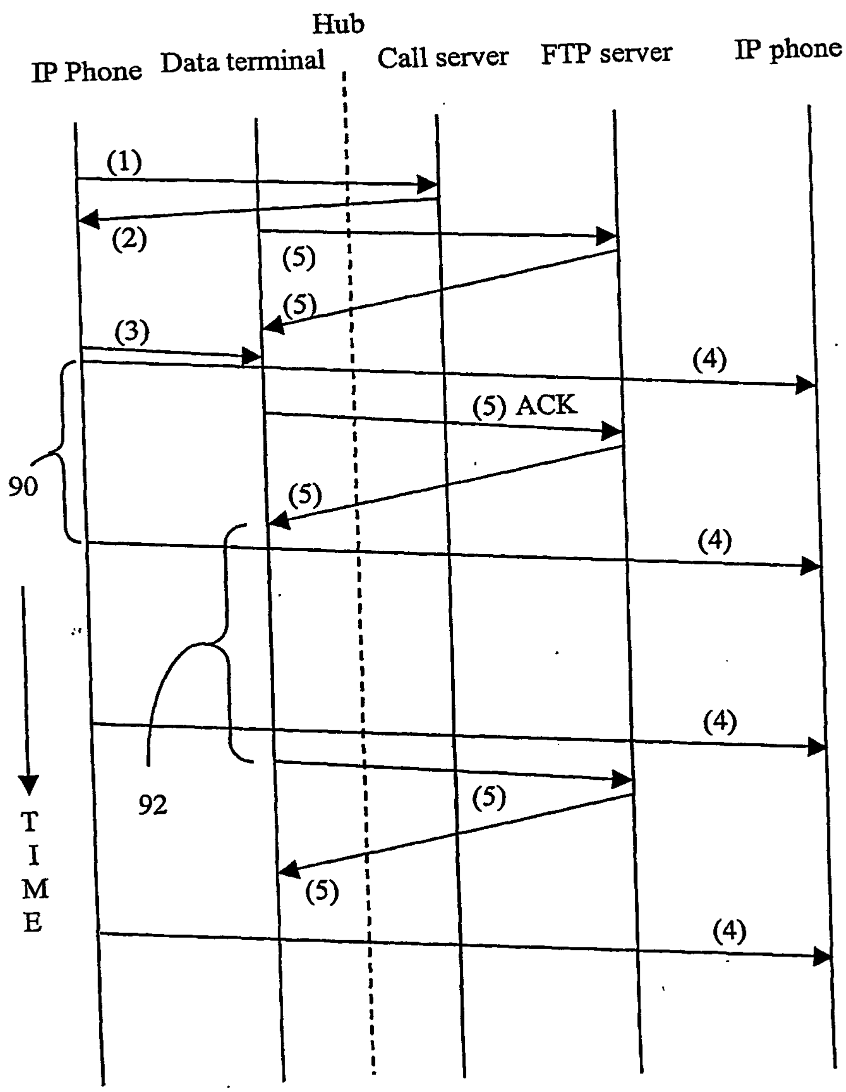

Hub
IP Phone
Data terminal
Call server
FTP server
IP phone
(1)
(2)
(5)
(5)
(3)
(4)
(5) ACK
90
(5)
(4)
(4)
92
(5)
(5)
(4)
TIME

Fig 8

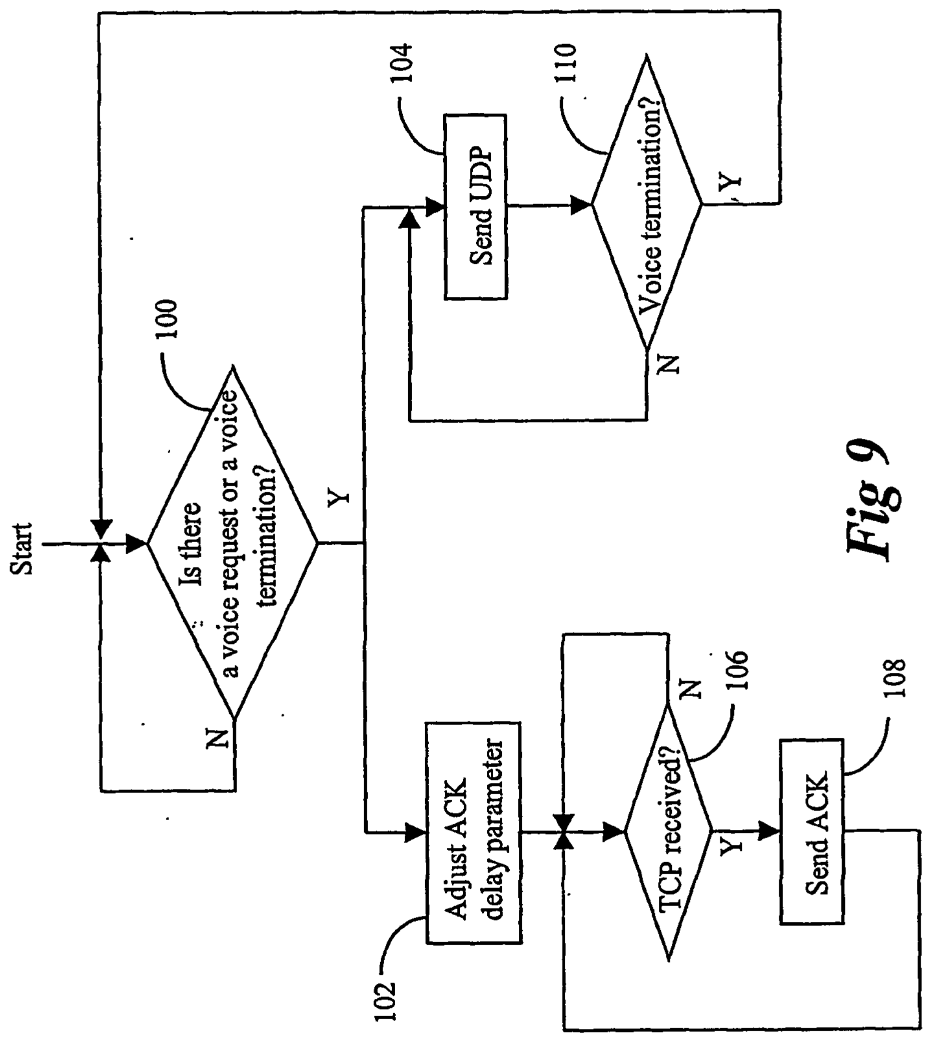

Start
Is there a voice request or a voice termination?
100
N
Y
Adjust ACK delay parameter
102
Send UDP
104
Voice termination?
110
N
Y
TCP received?
106
N
Y
Send ACK
108

Fig 9

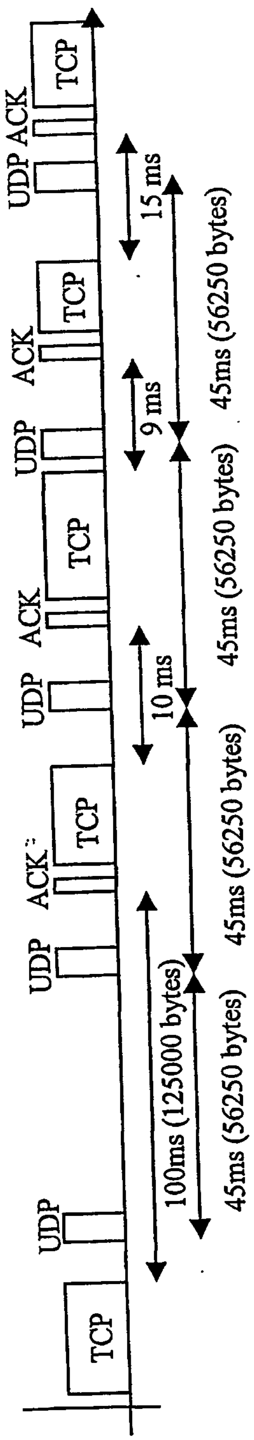

TCP
UDP ACK
15 ms
TCP
ACK
UDP
9 ms
TCP
ACK
UDP
10 ms
TCP
ACK
UDP
45ms (56250 bytes)
45ms (56250 bytes)
45ms (56250 bytes)
45ms (56250 bytes)
100ms (125000 bytes)
UDP
TCP

Fig 10

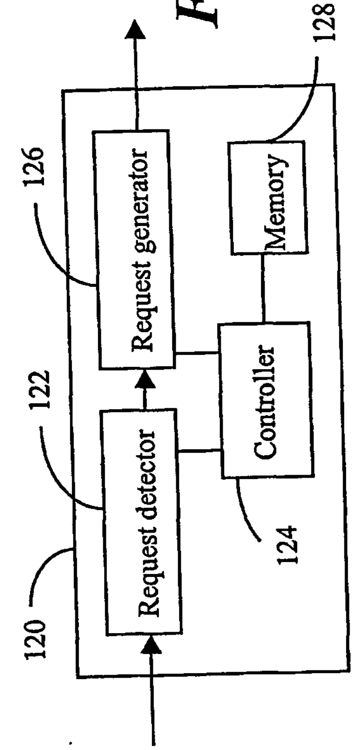

120
122
Request detector
126
Request generator
124
Controller
128
Memory

Fig 11

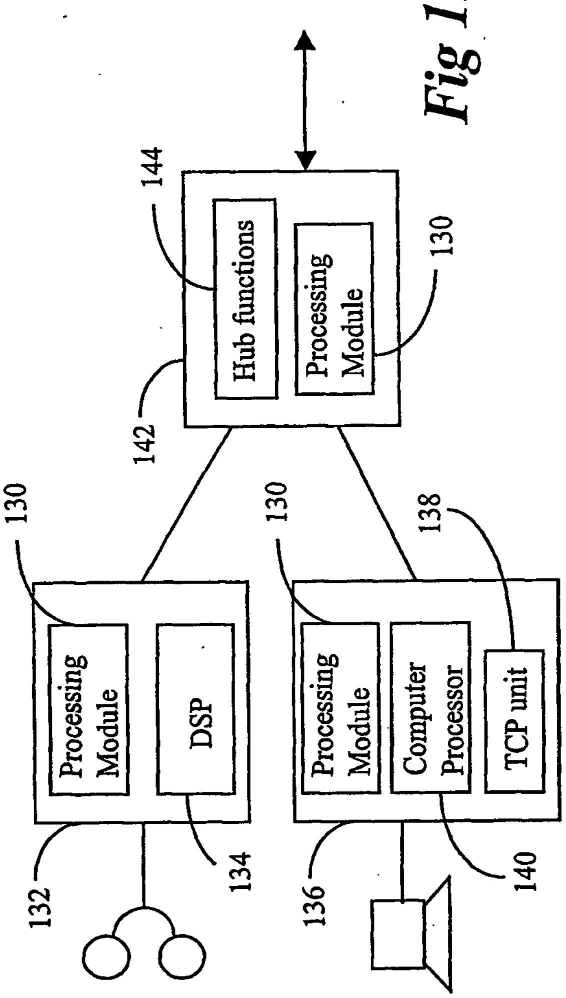

144
Hub functions
Processing Module
130
142
130
Processing Module
DSP
132
134
130
138
Processing Module
Computer Processor
TCP unit
136
140

Fig 12

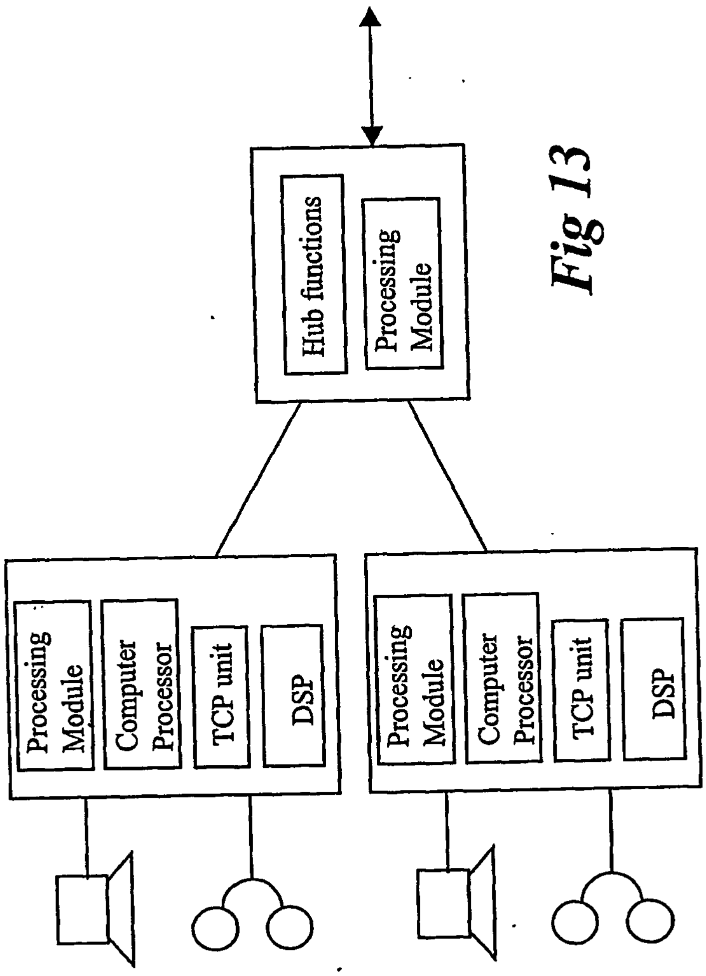

Hub functions
Processing Module
Processing Module
Computer Processor
TCP unit
DSP
Processing Module
Computer Processor
TCP unit
DSP

Fig 13

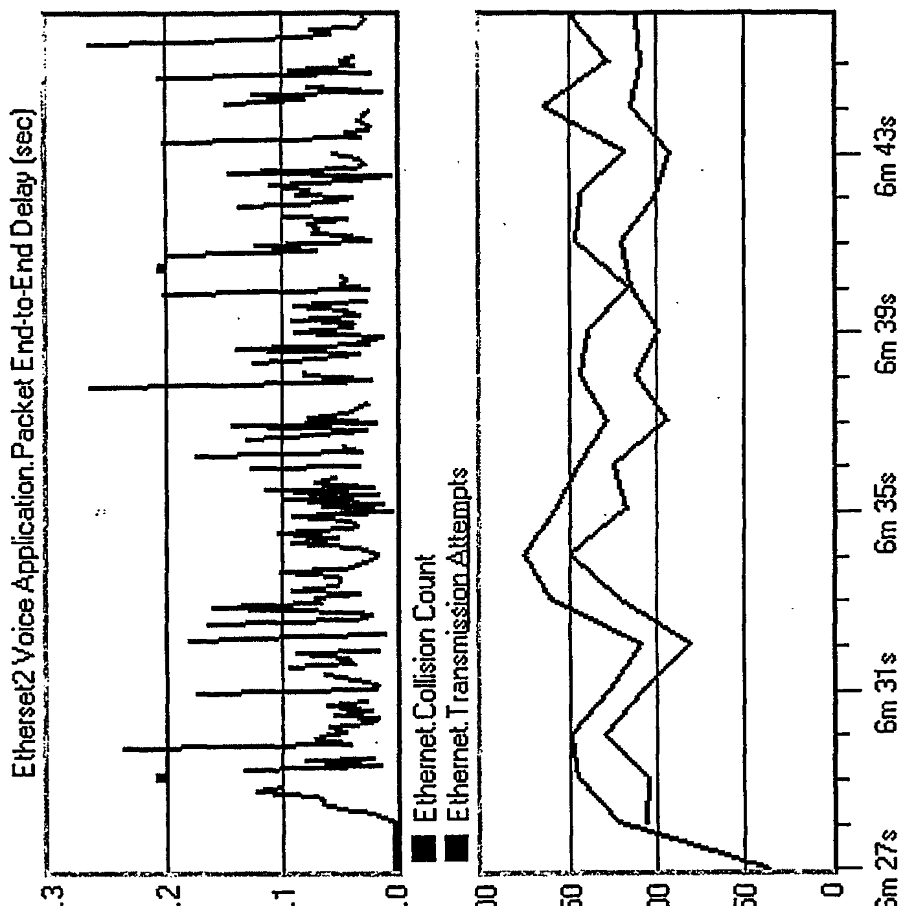

Ethernet2 Voice Application.Packet End-to-End Delay (sec)
0.3
0.2
0.1
0.0
Ethernet.Collision Count
Ethernet.Transmission Attempts
200
150
100
50
0
... 6m 27s
6m 31s
6m 35s
6m 39s
6m 43s

Fig 14

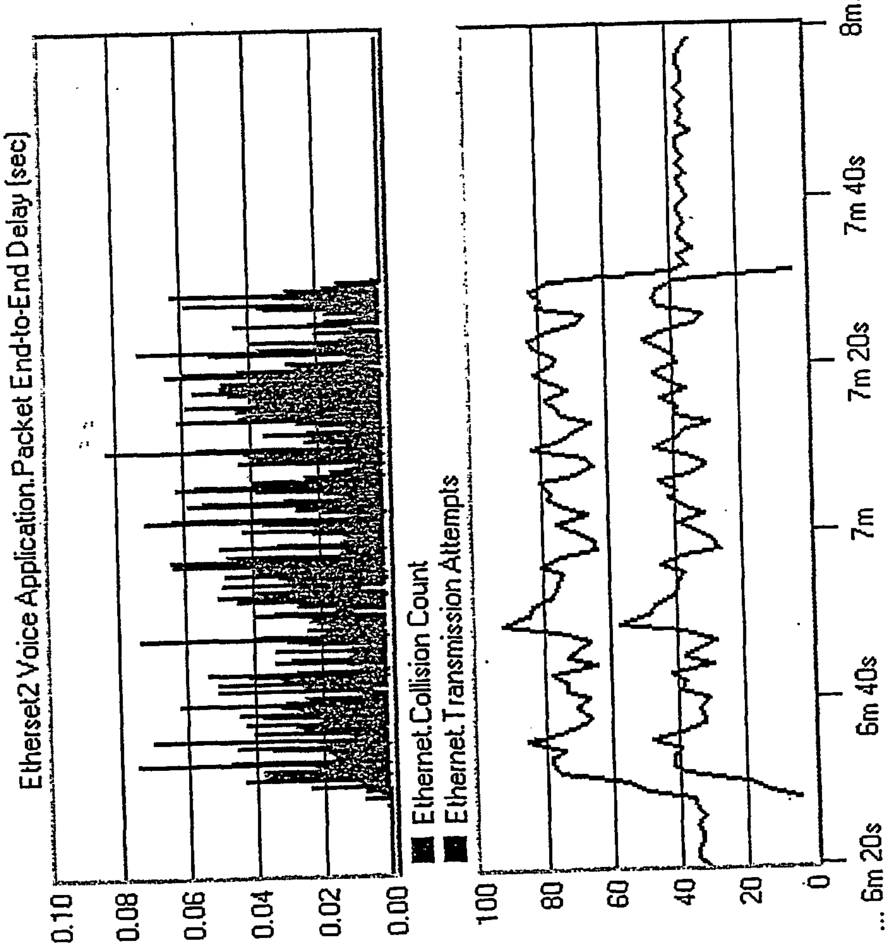

Ethersel2 Voice Application.Packet End-to-End Delay (sec)
0.10
0.08
0.06
0.04
0.02
0.00
Ethernet.Collision Count
Ethernet.Transmission Attempts
100
80
60
40
20
0
... 6m 20s
6m 40s
7m
7m 20s
7m 40s
8m..

Fig 15

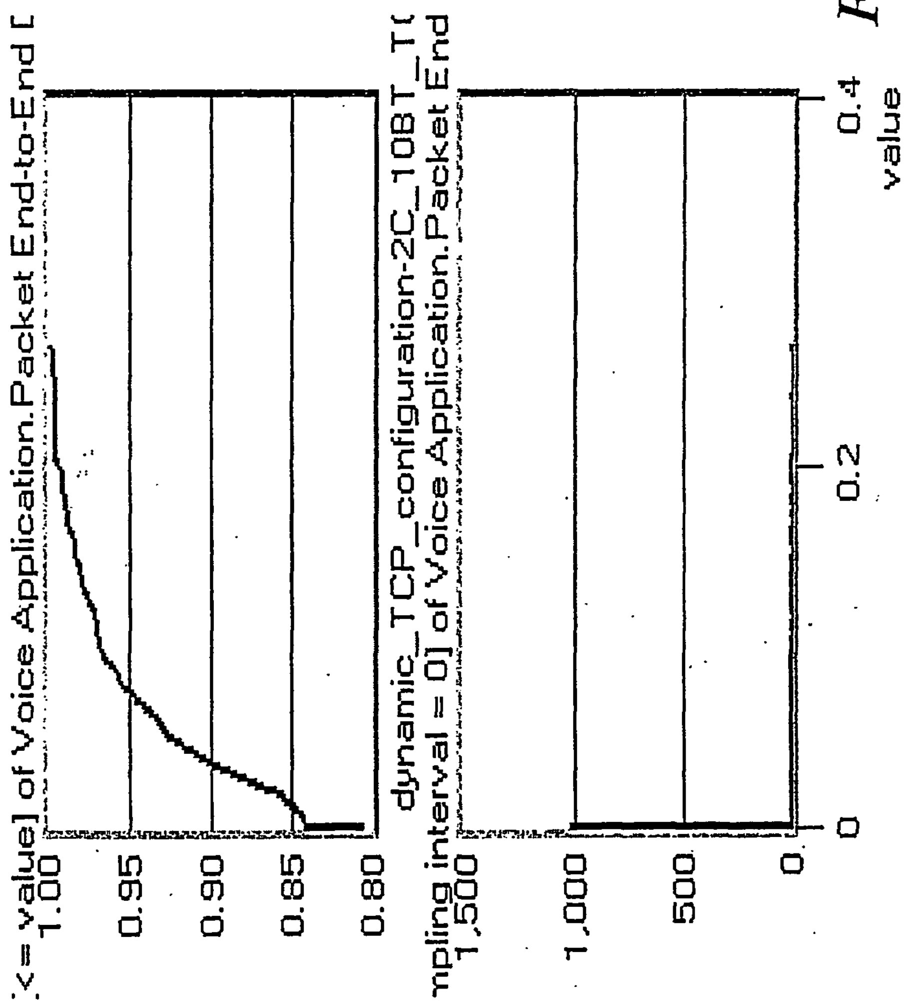

[<= value] of Voice Application.Packet End-to-End D
1.00
0.95
0.90
0.85
0.80
dynamic_TCP_configuration-2C_10BT_T
mpling interval = 0] of Voice Application.Packet End
1,500
1,000
500
0
0
0.2
0.4
value

Fig 16

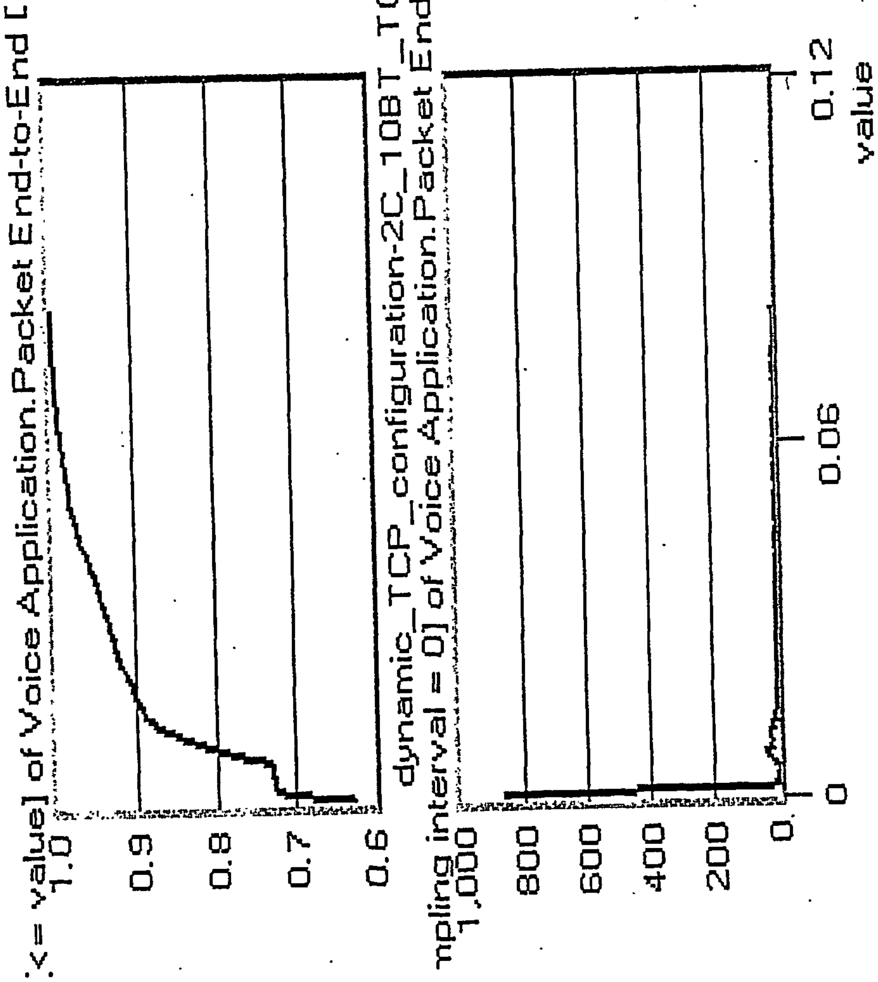

x <= value] of Voice Application.Packet End-to-End D
1.0
0.9
0.8
0.7
0.6
dynamic_TCP_configuration-2C_10BT_T
mpling interval = 0] of Voice Application.Packet End
1,000
800
600
400
200
0
0
0.06
0.12
value

Fig 17

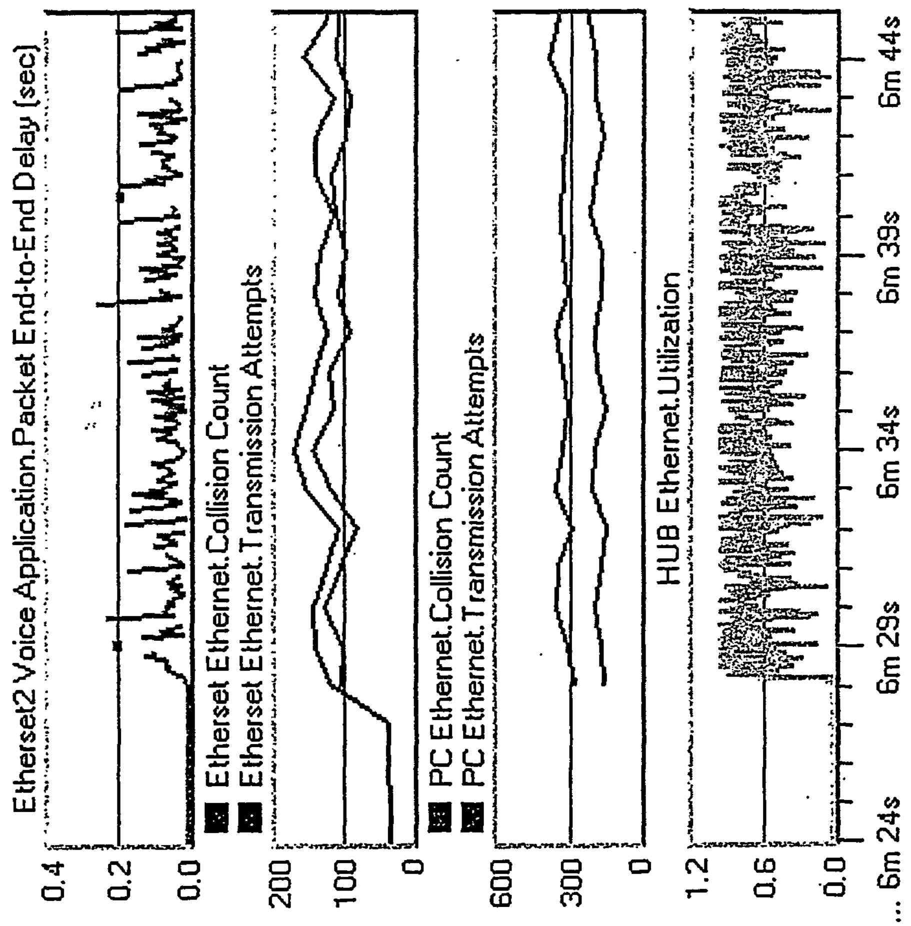

Etherset2 Voice Application.Packet End-to-End Delay (sec)
0.4
0.2
0.0
Etherset Ethernet.Collision Count
Etherset Ethernet.Transmission Attempts
200
100
0
PC Ethernet.Collision Count
PC Ethernet.Transmission Attempts
600
300
0
HUB Ethernet.Utilization
1.2
0.6
0.0
... 6m 24s
6m 29s
6m 34s
6m 39s
6m 44s

Fig 18

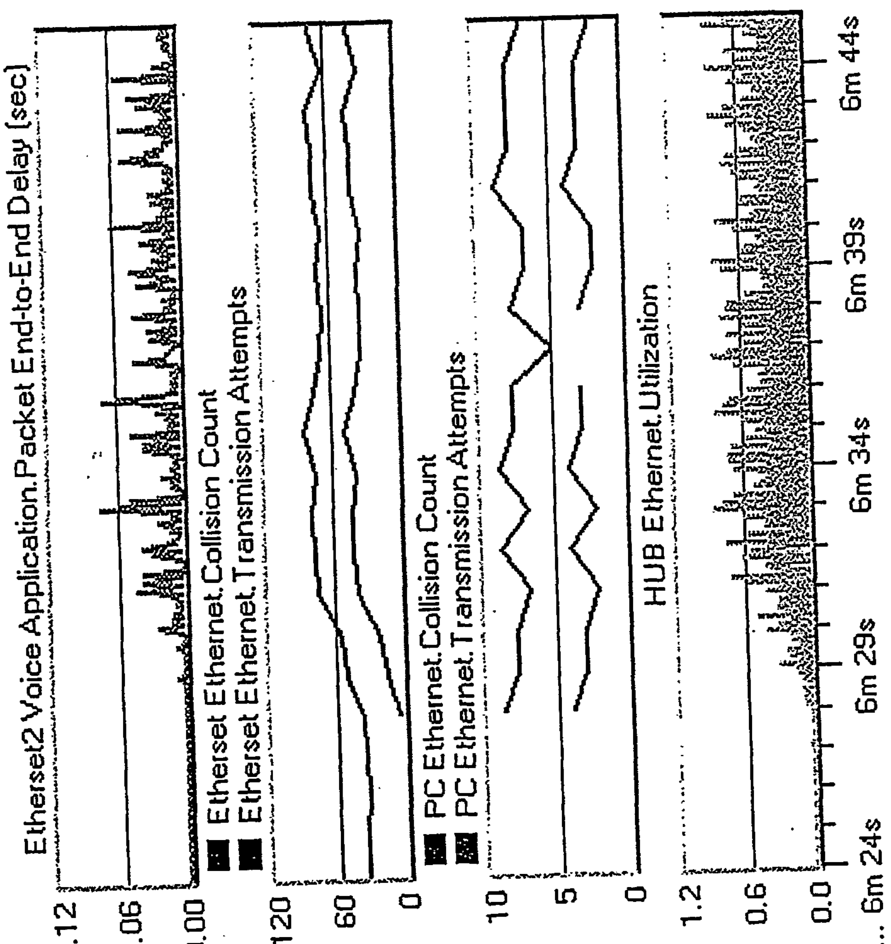

Etherset2 Voice Application.Packet End-to-End Delay (sec)
0.12
0.06
0.00
Etherset Ethernet.Collision Count
Etherset Ethernet.Transmission Attempts
120
60
0
PC Ethernet.Collision Count
PC Ethernet.Transmission Attempts
10
5
0
HUB Ethernet.Utilization
1.2
0.6
0.0
... 6m 24s
6m 29s
6m 34s
6m 39s
6m 44s

Fig 19